

## ASSIGNMENT 43.1 – SCALA 1

Mar 2018 batch - Student: K. Anandaranga

---

### 1. Task 1

- Download and import AcadgildSpark VM in Oracle Virtual box

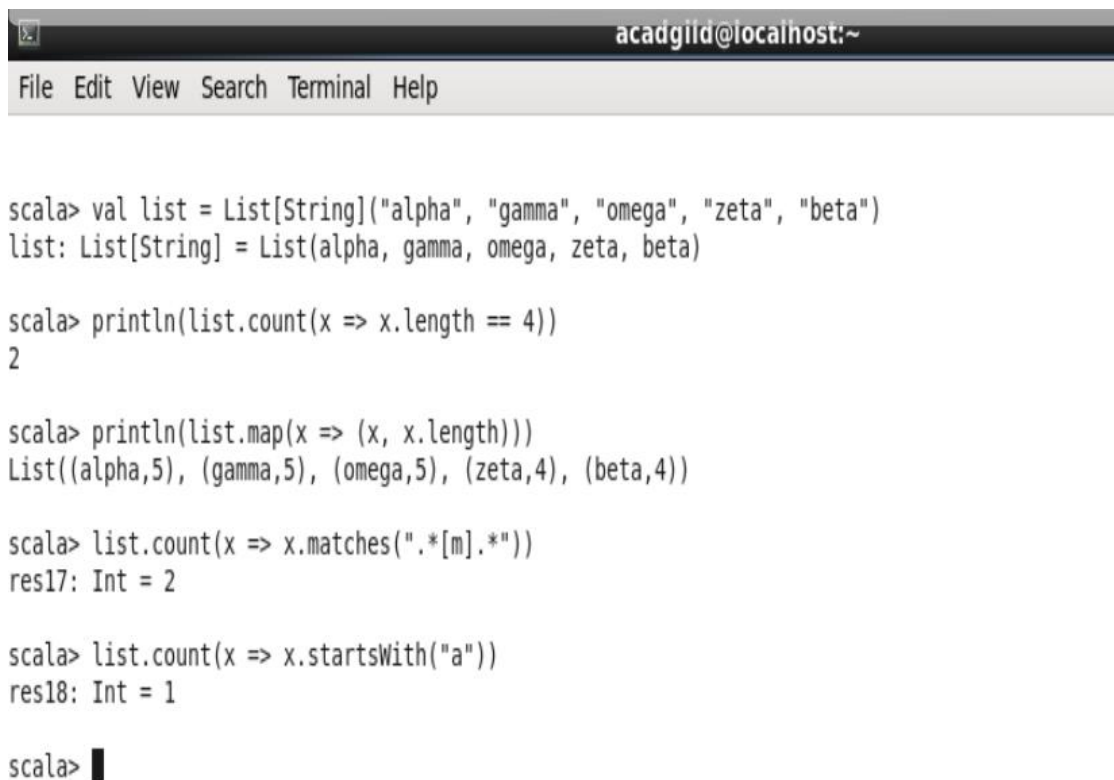
**Solution:** Successfully downloaded and installed the required software for 64-bit. Completed the assignments using the same.

### 2. Task 2

Given a list of strings - List[String] ("alpha", "gamma", "omega", "zeta", "beta")

- find count of all strings with length 4
- convert the list of string to a list of integers, where each string is mapped to its corresponding length
- find count of all strings which contain alphabet 'm'
- find the count of all strings which start with the alphabet 'a'

#### Solution:

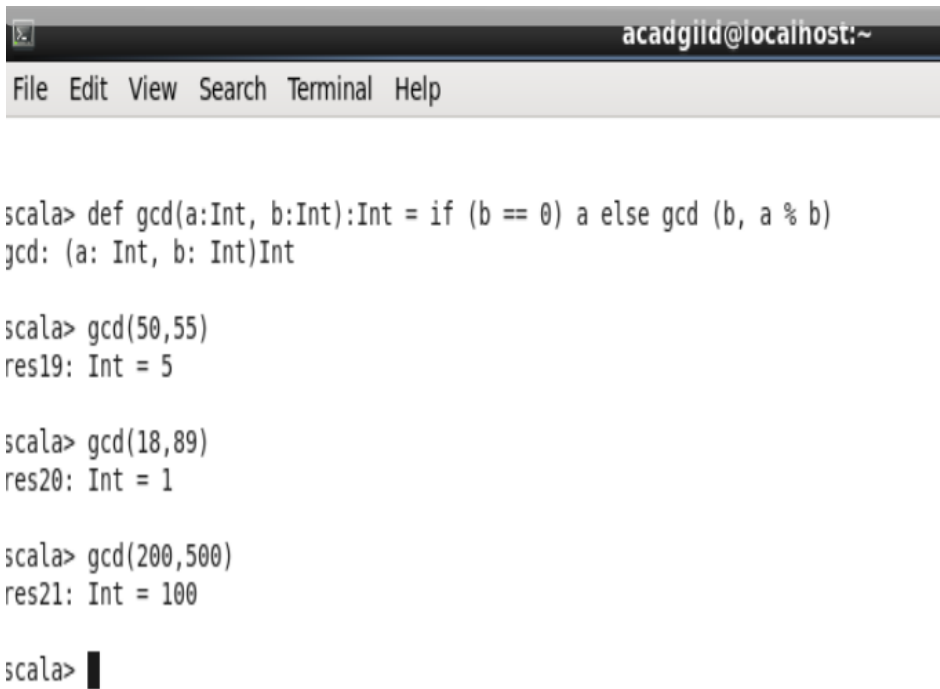


```
acadgild@localhost:~  
File Edit View Search Terminal Help  
  
scala> val list = List[String]("alpha", "gamma", "omega", "zeta", "beta")  
list: List[String] = List(alpha, gamma, omega, zeta, beta)  
  
scala> println(list.count(x => x.length == 4))  
2  
  
scala> println(list.map(x => (x, x.length)))  
List((alpha,5), (gamma,5), (omega,5), (zeta,4), (beta,4))  
  
scala> list.count(x => x.matches(".*[m].*"))  
res17: Int = 2  
  
scala> list.count(x => x.startsWith("a"))  
res18: Int = 1  
  
scala> █
```

### 3. Task 3

Create a Scala application to find the GCD of two numbers.

**Solution:**

A terminal window with a dark title bar containing the text 'acadgild@localhost:~'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows a Scala REPL session where a recursive GCD function is defined and then called with three different pairs of numbers: (50, 55), (18, 89), and (200, 500). The results are displayed as 'res19: Int = 5', 'res20: Int = 1', and 'res21: Int = 100' respectively. The prompt 'scala>' is followed by a black cursor bar.

```
acadgild@localhost:~  
File Edit View Search Terminal Help  
  
scala> def gcd(a:Int, b:Int):Int = if (b == 0) a else gcd (b, a % b)  
gcd: (a: Int, b: Int)Int  
  
scala> gcd(50,55)  
res19: Int = 5  
  
scala> gcd(18,89)  
res20: Int = 1  
  
scala> gcd(200,500)  
res21: Int = 100  
  
scala> █
```